Exploring Load Balancing To Solve Various Problems In Distributed Computing

Priyesh Kanungo[1]
Professor and Senior Systems Engineer (Computer Centre)
School of Computer Science and Information Technology
Devi Ahilya University
Indore-452001, India[1]

***Abstract*— Web base applications have grown huge and complex owing to dynamic computations and large number of dispersed users. This paper highlights some challenges before the researchers causing overload in the web based applications and necessitate the use of scheduling techniques like Dynamic Load Balancing (DLB) in Information Technology. The primary issues include mismatch of servers, lack of optimization algorithms in routers and heterogeneity of end servers. These issues make us aware of possible problems which may be encountered in distributed computing environment and provide a new direction to research in the area of distributed scheduling. We feel that such problems have been ignored by the research community and need urgent attention.**

***Keywords*—Optimization Algorithms in Routers, Network of Workstations, Computational Grid, Server Cluster, Load Balancing, Cloud Computing.**

I. INTRODUCTION

Distributed computing systems virtually combine geographically dispersed computing resources to satisfy the demand of compute intensive jobs. Network of Workstations (NOW), Computing Cluster, Server Farm, Computing Grid, P2P Computing, Cloud Computing etc. are some of the examples of distributed computing environment. NOW consists of loosely coupled systems of nodes and workstations and that can use idle resources of existing nodes when primary users are not using them. It provides cost effective solution to small computational problems. When the nodes in NOW are dedicated to particular application, it is called Beowolf cluster. Cluster Computing is an example of distributed computing system in which homogeneous nodes are interconnected through high speed network to process jobs with high computing needs. Clusters provide cost effective and fail safe alternative to mainframe and supercomputing model. Server farm or server cluster is a collection of replicated servers connected via high speed LAN [6].

A computational grid, as shown in Fig.1, consists of a set of clusters that are interconnected by gates. Each cluster may contain several sites that are connected by switches. A site consists of a number of processing elements as well as storage devices interconnected over a local area network. Grid infrastructure enables integrated collaborative use of high performance systems, networks, databases and variety of end user devices that are owned and managed by multiple organizations [3]. The objective is to provide computing utilities in the same manner as power utilities supply electric power. Similar objective was also dreamed way back in 1960s by means of MULTICS operating system project by the giants of that era viz. General Electric Company, MIT and Bell Labs. MULTICS was based on timesharing concept, but the dream could not be realized. Now, it has again been predicted that within next few years, users will draw their computing and storage power from grids and clouds rather than from local resources. Grid computing has been envisaged as the next revolution after WWW. A number of scientific and commercial applications have started harnessing grids [4].

Major issues in grid computing software include workload management and efficient utilization of resources by improved distributed scheduling techniques which fall in the scope of load balancing. Heterogeneity due to involvement of variety of resources also demands implementation of load balancing in the grid environment [8; 12]. The integration of load balancing in grid middleware software is shown in Figure 2.

Cloud computing addresses the next evolutionary step of distributed computing after cluster, P2P and Grid Computing and has emerged as a popular computing model to support large volumetric data using cluster of commodity computers. It is a promising paradigm for delivering various IT-Based services as computing utilities by making a better use of distributed resources to achieve better throughput and tackle large scale problems e.g. with exabytes (EB) of data. They are basically designed to provide the services to external users. The goal of this computing model is to make a better use of distributed resources to achieve higher throughput and tackle large scale computational problem. In clouds everything is a service, XaaS (anything as a service) where X can be Infrastructure, Platform, Software, Hardware, Business, Organization, Database, Desktop, Development etc. [9].

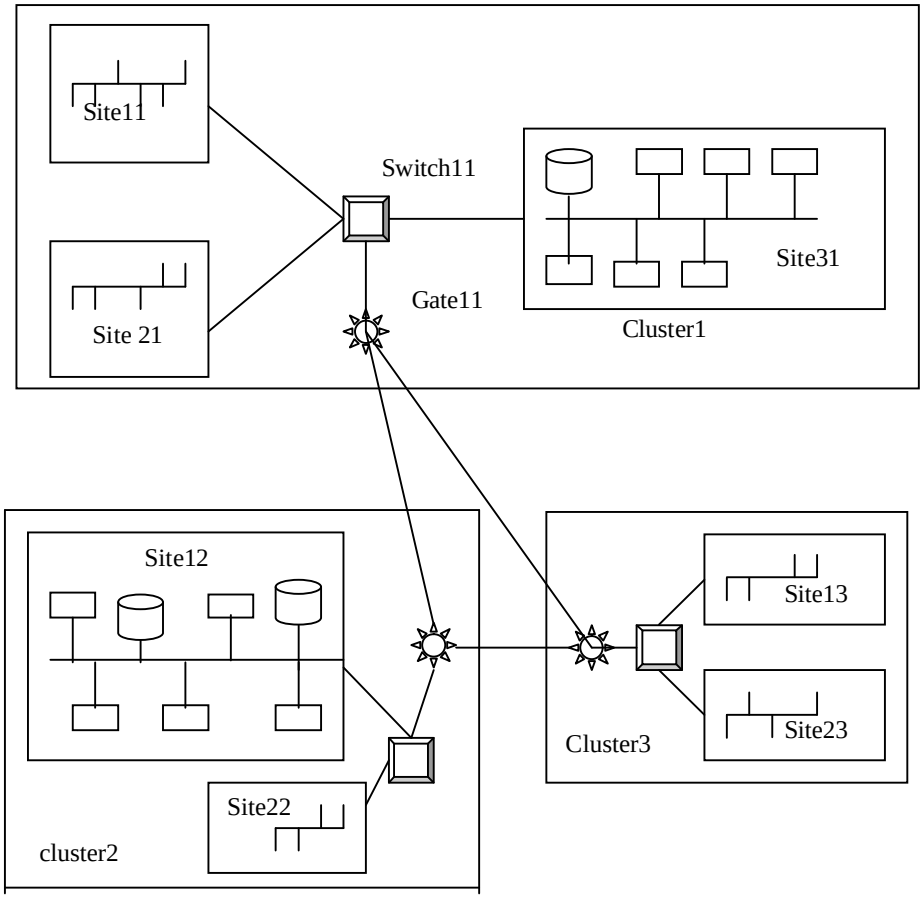

Fig. 1 A Computational Grid

II. Mismatch / Incompatibility of Severs

Breakthrough in software productivity depends on our ability to combine pieces of hardware and software to produce new applications. To make large and quality software rapidly, we need assembling of reusable building blocks into a new system rather than build-from-scratch techniques. Problems in composition are due to low-level issues of interoperability e.g. mismatches in programming languages, database schemas, operating platforms and other components on servers. Server mismatch exposes some fundamental problems and suggests possible research avenues to solve them. There may be conflicting assumptions about protocols, data models for RPC, topology, the type of user interfaces etc [12].

In addition to above components, many intermediaries are also supported for WWW. Intermediaries are the software entities deployed on Internet for flow of information from clients to the servers, among clients and among servers to cope with end user information overloading, support for mobile access, content personalization and infrastructural support for ubiquitous services like personal digital assistants (PDAs), thin clients, mobile phones etc. They are used for caching, filtering, indexing and transconding. The server mismatch problem is expected to grow further with the increasing application of these intermediaries.

High performance is one of the critical requirements demanded by mission critical applications such as finance, manufacturing etc. which are running on server clusters. Mismatch between OS and distinct characteristics of server applications is key performance bottleneck. Although servers have become very powerful with huge memory and hard disk space, it may be difficult to achieve satisfactory level of performance even after extensive system tuning efforts.

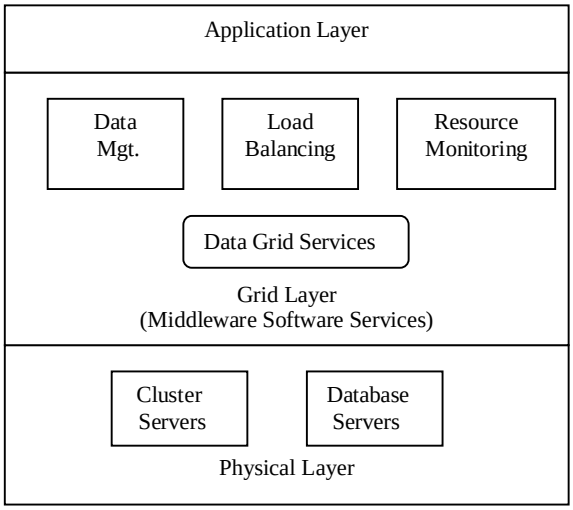

Fig. 2 Application of load balancing in grid architecture

In the process management for large server clusters, the major issue is scheduling of thousands of processes. For effectively scheduling such a large number of processes, intelligent scheduling decisions are desirable, e.g. pre-emptive processor scheduling may degrade the performance of some applications. One example is of database applications, which uses an efficient mutual exclusion technique called latches to synchronize access to shared data. If a process holding a latch is pre-empted by round robin scheduler, the process will join at the end of the ready queue. It is probable that the next process scheduled to run may also require latches, and therefore not able to use its time quantum. For taking intelligent decision, the scheduler must have application dependent information. Scheduler needs to have semantic information about applications and their execution states. User level threads can be also be used to avoid unnecessary context switches inside the kernel. User level threads are provided by a thread package, which allows users to implement their own algorithm to the schedule the threads within the allocated time quantum. Even if a thread executes a system call, other thread in the process can continue to execute [11]. Load balancing can be very useful in for scheduling processes across the servers in a cluster. Appropriate admission control and request distribution mechanism can be used for processing requests efficiently. Content aware scheduling algorithms for distributing client requests can be helpful in taking intelligent scheduling decisions depending on the type and capabilities of the servers [8].

III. Lack of Optimization Algorithm in Routers

Routers are used in heterogeneous networks to interconnect two different technologies and forward data packets between them. Routing is the act of moving information across the internetwork from a source node to the destination node. Along the way, at least one node is typically encountered. Routers determine the optimal routing path in the network and transport the packets through an internetwork called packet switching network. Routing protocols use metrics to evaluate what path will be best for a packet to travel. Different routers may use different matrices like path length, load on network resources, bandwidth, routing delay, reliability of network, communication cost etc. To aid to the process of path determination, routing algorithm uses a routing table which contains some routing information about the next hop or destination address. It may also contain other information like desirability of a path. When router receives an incoming packet, it checks the destination address and attempts to associate this address with the next hop. Operations in a router are depicted by means of Figure 3.

Routers communicate with one another and update their routing tables through transmission of variety of messages. The router update message is one such messages that consists of all or portion of routing table. By analysing routing update messages from all other routers, a router can build a detailed picture of network topology. Another message, called link state advertisement, informs other routers about the state of the sender's links. This message can also be used to build complete picture of the network topology to determine the optimal route of a network destination. Switching algorithms are also used in routers and are generally same for most routing protocols. In most of the cases, a host determines router address and sends packet to this address along with the protocol address of the destination host. The next hop may not be ultimate destination host but again a router which executes the same switching process. As the packet moves through the network, its physical address changes but its protocol address remains the same [10].

Various routing algorithms exist and each algorithm has a different impact on network and router resources. Optimality is the most desirable feature of routing algorithms. It is the capability of a routing algorithm to select best route to transfer a packet. The algorithm uses matrix weighing to make calculations. Some algorithms use number of hops and delays. Weight given to different matrices may also vary. One algorithm may give more weight to the number of hops while another algorithm gives weight to delay. The algorithm should also be able to converge rapidly. Convergence is the process of agreement by all other routers on optimal path. Most of the router algorithms use dynamic routing i.e. they adjust to changing network circumstances by analysing incoming routing update messages. If a message indicates that network change has occurred, the algorithm recalculates the route and sends new routing update message. These messages permeate through the network, stimulating routers to re-execute their algorithms and change their routing tables accordingly. Some of the sophisticated routing protocols support multiple paths to the same destination to allow traffic multiplexing over multiple lines to improve reliability and throughput[2].

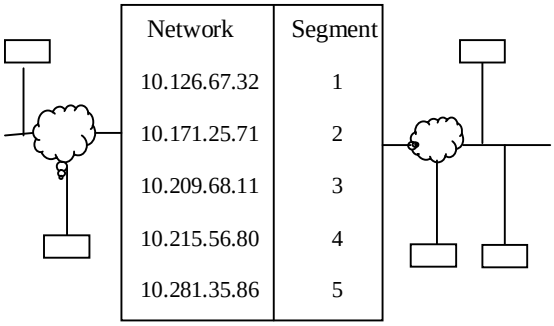

Fig 3. Operation of the router in a network

Internet bandwidth explosion and advent of complex distributed applications had presented new challenges for routers. These challenges include more throughput, more computations and more flexibility. In minimal routing algorithms, which try to choose shortest path for each packet, load imbalance may be caused due to heavy load on some links and less load on other [11].

Load balancing can be implemented at OS level, at middleware level or at the network level. Routers help to achieve load balancing at network level. To improve worst case traffic, routing algorithms must balance load by sending some of the load over non-optimal paths also. Load balancing can be incorporated in routers to distribute traffic over all the router ports that are at the same distance from destination address. Load balancing increases utilization of network segments and consequently effective network bandwidth. Load balancing algorithms can be implemented in many ways in routers and at various levels of networking protocol stack [10].

IV. Performance and Heterogeneity of End Servers

In a client server environment, it is common to have a cluster of replicated servers which accepts processing requests from the large number of clients. A cluster is a group of servers with identical contents, networked together to act as a single virtual server and capable of growing with corporate needs [12].

For some organizations, a few servers and storage devices are sufficient which are easily manageable. But the organizations that support wider spectrum of applications and requirements have developed a framework where capability of server is matched to the type of application. This is called n-tier architecture. The first tier contains external interface for request processing, supporting Internet applications such as web servers, firewalls, caching mechanism etc. The server architecture can be relatively simple. These applications do not require much integration with rest of the infrastructure. The second tier of servers contains application specific servers supporting mission critical applications. The servers require greater functionality to support varying demands from applications. Often scale up server configuration may be required. Number of processors is usually more and system performance between input/output and computations has to be balanced. Applications may be highly interdependent and may require better tuning compared to first tier applications.

The third and final tier is database layer where large servers are needed for sophisticated database products. These servers are multiprocessor-based with rich functionality due to extensive processing needs e.g. online transaction processing (OLTP), data warehousing etc. Figure 4 shows a three-tier structure of a server cluster [5]. Heterogeneity within group of servers used in three-tier architecture of a server cluster raises the problem of how to distribute clients' requests to the different cluster nodes. The workload consisting of incoming requests is to be distributed evenly among the servers in the cluster [1]. For a heterogeneous server cluster to achieve its high performance and high availability potential, DLB techniques are required. Combining load balancing with cluster of low cost servers is most cost effective, flexible and reliable strategy for web based services. Cluster load management software should efficiently support heterogeneous hardware environment and changing systems configurations. .

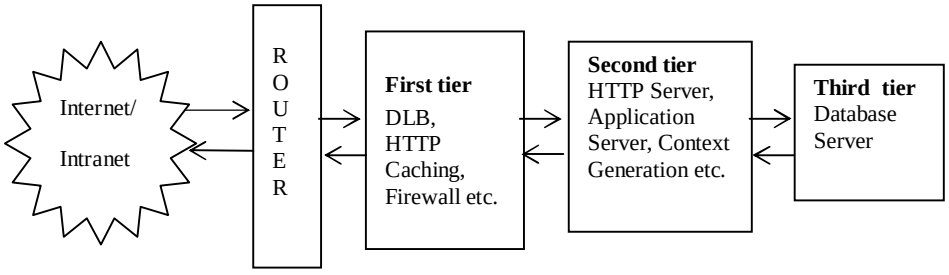

Fig. 4. Three-tier architecture of a server cluster

Distribution of requests among servers can be implemented by monitoring the servers regularly and directing these requests dynamically to the least loaded servers. The capability of servers has to be taken into account while distributing the requests. DLB feature can be added to the pre-existing domain name service as it plays a crucial role in resolving client requests. DLB optimizes request distribution among servers based on factors like server capacity, current load and historical performance. It also improves mean response time and overall throughput of a DCS [11].

V. CONCLUSIONS

DLB is an important research theme owing to the development of ever growing distributed computing applications. Research community has given considerable attention to developing optimal and efficient techniques of DLB for various distributed applications. There is a great deal of research scope in solving these problems using DLB methodologies. The areas described above pose challenges for further research and provide us opportunities to come up with new ideas.

In this paper, we have identified challenges before the researchers, which contribute to the growth of Internet traffic and have made the use of load balancing necessary in Information Technology applications. We have discussed these issues and seen how DLB concept can be utilized to enhance the performance. The methodologies and algorithms developed in the paper are found to be useful in meeting these new challenges face by distributed computing environment.

REFERENCES

[1] M. Aron, P. Druschel and W. Zwaenepoel, "Cluster Reserves: A Mechanism for Resource Management in Cluster-Based Network Servers," *Proceedings of the ACM SIGMETRICS, International Conference on Measurement and Modeling of computer Systems*, Santa Clara, CA, Vol. 28, pp. 99-101, June 2000.

[2] L. Aversa and A. Bestavros, "Load Balancing a Cluster of Servers using Distributed Packet Rewriting," *Proceeding of the IEEE International Performance, Computing and Communications Conference*, Pheonix U.S.A, pp. 24-29., Feb. 2000.

[3] M. Baker, R. Bhuyy and D. Laforenz, "Grids and Grid Technologies for Wide-Area Distributed Computing," *International Journal of Software: Practice and Experience (SPE)*, Vol. 32, No. 15, pp. 1437-1466, 2002.

[4] I. Foster, C. Kesselman and S. Tuecke, "Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of High Performance Computing Applications*, Vol. 15, No. 3, pp. 200-222, 2001.

[5] J.O. Kephart and D.M. Chess, "The Vision of Automatic Computing," *IEEE Computer Magazine*, Vol. 36, No. 1, pp. 41-50, 2003.

[6] H. Mehta, P. Kanungo and M. Chandwani, "Performance Enhancement of Scheduling Algorithms in Clusters and Grids using Improved Dynamic Load Balancing Techniques," 20th International World Wide Web Conference 2011 (PhD Symposium), Hosted by IIIT, Banglore at Hyderabad, 28 March-01 April 2011, pp. 385-389, Awarded NIXI (National Internet Exchange of India) Fellowship.

[7] W. Scacchi, "Understanding the Requirements for Developing Open Source Software Systems," *IEEE Proceedings–Software*, Vol. 149, No. 1, 2001.

[8] R K Sharma, P. Kanungo and M. Chandwani, "A New Dynamic Load Balancing Algorithm Based on Workstation Priority in Network of Workstations using MPI Environment," Journal of Institution of Engineers, Computer Engineering, Vol. 92, 2011

[9] R K Sharma, P. Kanungo and M. Chandwani, "A Green Cloud Computing Architecture Supporting e-Governance," International Conference on Automation and Computing, University of Huddersfield, Huddersfield, United Kingdom, Sept. 2011.

[10] W. Shi, M.H. MacGreger and P Gburzynski, "Load Balancing for Parallel Forwarding," *IEEE/ACM Transactions on Networking*, Vol. 13, No. 4, pp. 790- 801, Aug. 2005.

[11] A. Tiwari and P. Kanungo, "Dynamic Load Balancing Algorithm for Scalable Heterogeneous Web Server Cluster with Content Awareness," 2nd International Conference on Trendz in Information Sciences & Computing, (TISC), Satyabhama University, Chennai, India, pp. 143-148, 2010.

[12] B. Yagoubi and Y. Slimani, "Dynamic Load Balancing Strategy for Grid Computing," *Enformatica Transactions on Engineering Computing and Technology*, Vol. 13, pp. 260-265, 2006.